Final Report

A Linguistic Foundation for Communicating Geo-Information in the context of BML and geoBML

by

Prof. Dr. Ulrich Schade
Dr. Miłosław Frey

Fraunhofer-FKIE
Neuenahrer Straße 20
53343 Wachtberg
GERMANY

to

US Army Engineer Research & Development
Center - International Research Office
Tel: +44-(0)-1895-61-6109

under

Contract number: W911NF-08-1-0456

23 March 2010

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
27-03-2010

2. REPORT TYPE
Final Report

3. DATES COVERED *(From - To)*
28-09-2008 to 27-03-2010

4. TITLE AND SUBTITLE
A Linguistic Foundation for Communicating Geo-Information in the context of BML and geoBML

5a. CONTRACT NUMBER
W911NF-08-1-0456

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. Ulrich Schade
Dr. Miłosław Frey

5d. PROJECT NUMBER
W90C2K1271EN01

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Fraunhofer-FKIE
Neuenahrer Straße 20
53343 Wachtberg
GERMANY

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
ERDC European Research Office
Edison House,
223 Old Marylebone Road
London NW1 5TH, UK

10. SPONSOR/MONITOR'S ACRONYM(S)
ERDC - ERO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
None

12. DISTRIBUTION / AVAILABILITY STATEMENT
Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Battle Management Language (BML) is a formal language for military communication, i.e., exchanging orders and reports. Orders and reports refer to geo-information. In particular, orders use overlays to connect geo-information to task assignments. The problem studied by this project is how to formalize geo-information and overlays, and to connect this formalized information to BML orders and reports such that a system can process them coherently and operate on the processed orders and reports in the way intended.

BML expressions can be generated by the Command and Control Lexical Grammar (C2LG) that is based on linguistic principles (i.e., lexicality, principle of coherence, principle of completeness). In the project, the grammar has been broadened. In addition, formal representations of tactical spatial objects and of overlays have been developed that, on the one hand, are XML representations and, on the other hand, can be connected to C2LG expressions, such that the expressions can be parsed to argument structures which include both the task assignment information and the corresponding geo-information. To preserve the linguistic principles, so-called overlay excerpts have been defined and developed. These excerpts ensure that only those parts of an overlay are connected to a task assignment which are of relevance for that assignment. The central elements of these overlay excerpts (and thus also the elements of the overlays) are formal representations of tactical spatial objects (mostly control features). These tactical spatial objects derive from the geoBML process. In order to make them processable in connection with formal orders and reports, the formal representation of the tactical spatial objects also has been structured in accordance with the linguistic principles and in accordance with doctrine.

15. SUBJECT TERMS
Computational Linguistics, Lexical Functional Grammar, Machine Translation, Geospatial

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE
U	U	U

17. LIMITATION OF ABSTRACT
U

18. NUMBER OF PAGES

19a. NAME OF RESPONSIBLE PERSON
Dr. Ulrich Schade

19b. TELEPHONE NUMBER *(include area code)*
++49 228 9435376

Abstract

Battle Management Language (BML) is a formal language for military communication, i.e., exchanging orders and reports. Orders and reports refer to geo-information. In particular, orders use overlays to connect geo-information to task assignments. The problem studied by this project is how to formalize geo-information and overlays, and to connect this formalized information to BML orders and reports such that a system can process them coherently and operate on the processed orders and reports in the way intended.

BML expressions can be generated by the Command and Control Lexical Grammar (C2LG) that is based on linguistic principles (i.e., lexicality, principle of coherence, principle of completeness). In the project, the grammar has been broadened. In addition, formal representations of tactical spatial objects and of overlays have been developed that, on the one hand, are XML representations and, on the other hand, can be connected to C2LG expressions, such that the expressions can be parsed to argument structures which include both the task assignment information and the corresponding geo-information. To preserve the linguistic principles, so-called overlay excerpts have been defined and developed. These excerpts ensure that only those parts of an overlay are connected to a task assignment which are of relevance for that assignment. The central elements of these overlay excerpts (and thus also the elements of the overlays) are formal representations of tactical spatial objects (mostly control features). These tactical spatial objects derive from the geoBML process. In order to make them processable in connection with formal orders and reports, the formal representation of the tactical spatial objects also has been structured in accordance with the linguistic principles and in accordance with doctrine.

Contents

Introduction

This report describes the research and the results of project "A Linguistic Foundation for Communicating Geo-Information in the context of BML and geoBML" (September, 28th, 2008 to March, 20th, 2010). The research has been carried out by the authors (investigators at Fraunhofer FKIE, formerly FGAN-FKIE) in cooperation with Prof. Dr. Michael R. Hieb, Center of Excellence for C4I, George Mason University, Fairfax, USA, and in accordance with and under the guidance of Mr. Lloyd Hauck and Mr. Richard Tynes, both of the US Army Engineer Research and Development Center, Topographic Engineering Center, Alexandria, VA, USA.

The focus of the research was the development of a formal computational grammar that a) is coherent with the Command and Control Lexical Grammar (C2LG), a grammar defining a Battle Management Language (BML), b) allows the formal representation of overlays and to express geospatial information in unambiguous terms, and c) proposes a method and a process to connect the representations of overlays and tactical spatial objects to C2LG expressions.

Linguistic Foundation

The formal computational grammar to be developed will be based on the Command and Control Lexical Grammar (C2LG), a grammar for C2 languages, among them BML (Schade & Hieb, 2006a, 2006b, 2007). C2LG is modeled upon Lexical Functional Grammar (LFG) (Bresnan, 2001), one of the major formal grammars in Linguistics. LFG is especially used in the field of Computational Linguistics for Natural Language Processing tasks, e.g., for Machine Translation. Because BML is not a natural language, C2LG is somewhat less complex than LFG. The analysis of a sentence according to LFG consists of three steps. First, the "constituents" of the sentences are calculated. Constituents are the linguistics equivalents of the 5 Ws (Who, What, When, Where, Why). For example, in "*The unit approached the phase line*", "*the unit*" as well as the "*the phase line*" are constituents (forming the Who and the Where, respectively). In LFG, the structure built by the constituents is called c-structure (constituent structure). In a second step, LFG transforms the c-structure into the f-structure, the functional structure. Functional structures are built by attribute-value pairs like XML. In principle, in functional structures, syntactic labels like "subject" or "object" are assigned to the constituents. E.g., "*the unit*" of our example would receive the syntactic label "subject" under the f-structure. In the third step, the a-structure (the argument structure) is built. Here semantic roles (also called thematic roles, cf. Sowa, 2000) are assigned. E.g., "*the unit*" would get the role "agent" ("an active animate entity that voluntarily initiates the action", Sowa, 2000, p. 508). The assignment of semantic roles is necessary to allow an automatic semantic interpretation of an expression. However, it is a difficult task to assign these kinds of roles to natural language expressions and their constituents. E.g., in sentences in passive voice, the constituent with the syntactic label "subject" is not the agent of the action. In computational linguistics, the assignment of semantic roles to constituents has become a major field of research, called "Semantic Role Labeling" (Jurafsky & Martin, 2009).

Since BML is not a natural language, but a formal one defined by a formal grammar, in our case C2LG, difficulties that had complicated the definition and the development of LFG have been avoided. C2LG is developed in a specific way. The sequence of the constituents is rigid and key words are to be used in C2LG expressions. As a result, the assignment of semantic roles can be directly calculated from the constituent structure of a C2LG expression. Thus, the calculation of the intermediate f-structure, the main problem in analyzing natural language expressions by LFG, does not apply. This may raise the question of why model a BML grammar after LFG.

LFG incorporates central linguistic principles which are needed both for the C2LG as well as for a geoBML grammar that expands the C2LG. First, LFG is *lexically driven* as indicated by the "Lexical" in its name. This means that the (formal) lexicon of the language provides information for the calculations to construct the c-, f-, and a-structures. This is an especially valuable property for a BML in general and for its geospatial aspects in particular because the task determines what kinds of spatial objects and control features need to be referred to if the task is to be assigned to a unit appropriately. Thus, our grammar had been shaped in a way that the lexical element denoting the task determines what kind of constituents are in the expression as well as what kind of spatial objects and control features are to be associated with the task. This dependency on lexical information is further developed in LFG's principles of coherence and completeness.

The *principle of coherence* says that only those constituents are allowed in an expression which are licensed by the lexical entry of the expression's "head" (in a task assignment expression this is the verb denoting the task). Applied to the geospatial domain, this means, for example, that a "move to contact" task whose expression contains a screen line as control feature is not coherent since a move to contact does not "license" a screen line.

The *principle of completeness* says that all constituents demanded by an expression's head must be part of that expression. Applied to the geospatial domain, this means, for example, that a "move to contact" task whose expression does not contain a "limit of advance" as control feature is not complete since a move to contact demands such a line. In short, the principles inherited from LFG by the geospatial grammar guarantee that tasks will be assigned to units in a way that there will be references to exactly those control features needed.

The Geospatial Grammar

In the following, the geospatial grammar that has been developed is presented. This grammar is coherent with the C2LG. Therefore, this chapter starts with a section providing an overview of C2LG. C2LG follows doctrine as does the geospatial grammar. According to doctrine, overlays are attached to operation orders to provide additional information for a commander as to how to interpret the written part of the order in general and its geospatial aspects in particular. The central problem for automatic analysis of an order under the geospatial view is how to make these overlays interpretable for systems and how to connect the information within these overlays to the information in the written parts, especially in the task assignments in the third paragraph of the Operation Order. This is the problem the geospatial grammar solves. Thus, the second section will be about how to formally represent overlays so that these representations can be connected to C2LG expressions. The third section is about overlay excerpts. Overlay excerpts are defined with respect to the assignment of a task to a unit. The overlay excerpt contains all information from the general overlay which has to be known in order to execute the task as intended. The fourth section will be about the representation of tactical special objects serving as control features for task assignments. These objects, naturally, are elements of overlays but their representation has to be viewed in detail. The final section will present how an order, with an attached overlay, can be written in C2LG expressions and processed.

The Command and Control Lexical Grammar (C2LG)

In this section, the C2LG is presented. C2LG is a formal grammar. As such, it follows the definition of formal grammars in general as proposed by Chomsky (1957). According to that definition, a grammar is a quadruple, consisting of a starting symbol, a finite set of terminal symbols (the grammar's lexicon), a finite set of non-terminal symbols (describing the kind of constituents and sentence-equivalent expressions for which the grammar allows generation), and a finite set of production rules determining how to connect the words to constituents and sentence-equivalent expressions. With respect to Chomsky's definition, C2LG is a so-called "context-free grammar". This means, in short, that each C2LG rule has exactly one non-terminal symbol on its left side and a sequence of terminal and non-terminal symbols on the right side. For example, "StartWhen → ***start*** TemporalQualifier DateTimeValue" is a typical rule; it says that the non-terminal symbol "StartWhen" (the symbol of the left side) is expanded to a sequence that begins with the terminal symbol "***start***" and is followed by a temporal qualifier – e.g., "*not later than*" – as indicated by the non-terminal symbol "TemporalQualifier" and by a date-time expression as indicated by the non-terminal symbol "DateTimeValue".

The part of the C2LG which is relevant for this study is the part called "tasking grammar" (Schade & Hieb, 2006a). Other parts, for example, deal with reports (cf. Schade & Hieb, 2007), or the representation of intent (Hieb & Schade, 2007). The C2LG rules listed in the tasking grammar are for assigning tasks to units. Since C2LG is a lexical grammar, it includes one basic task assignment rule specific to each task in question. All these basic rules follow the format given in (1). Example rules for the tasks "advance", "assist", "block", "defend", and "march" are shown in (2a) to (2e). These examples are also presented in Schade & Hieb (2006a) with a more thorough discussion than provided here.

(1) OB → Verb Tasker Taskee (Affected|Action) Where
StartWhen (EndWhen) Mod Why Label

(2a) OB → ***advance*** Tasker Taskee RouteWhere
StartWhen (EndWhen) Mod Why Label

(2b) OB → ***assist*** Tasker Taskee Action AtWhere
StartWhen (EndWhen) Mod Why Label

(2c) OB → ***block*** Tasker Taskee Affected AtWhere
StartWhen (EndWhen) Mod Why Label

(2d) OB → ***defend*** Tasker Taskee Affected AtWhere
(EndWhen) Mod Why Label

(2e) OB → ***march*** Tasker Taskee RouteWhere
StartWhen (EndWhen) Mod Why Label

The rule format (1) expresses the sequence of the constituents in the task assignment expressions. First, there is a term that specifies the task to be assigned (Verb), e.g., "*advance*" or "*move to contact*". This is followed by the constituents denoting who assigns the task (Tasker) and to whom the task is assigned (Taskee). The non-terminal symbols Tasker and

Taskee expand to the names or the IDs of the respective units. The fourth constituent denotes what is affected by the task. It is either an object (Affected), such as an enemy unit affected by an ambush, or another action (Action). The specifics for this constituent are determined by the task in question, e.g., "*assist*" demands Action, "*block*" demands Affected, and "*advance*" prohibits these kinds of constituents. The assignment expressions are completed by a spatial constituent (Where), one or two temporal constituents (StartWhen, which is mandatory, and EndWhen, which is optional as indicated by the round brackets), a modifying constituent (Mod), a constituent to express the purpose of the assigned task (Why), and a label (Label). The label is used if the assigned task is to be referred to in another expression. An example of such an expression is provided in rule (3).

(3) [***task assignment***] *occupy BN-661 Coy2 Prins Willem-Alexander Brug at Parnass*
start at TP1 in-manner fast in-order-to enable label-o24 label-o23;

The task assignment (3) says that the battalion *BN-661* (the Tasker) orders its second company (*Coy2*, the Taskee) to occupy a bridge, the so-called "*Prins Willem-Alexander Brug*" (Affected). This is supposed to happen *at Parnass* (the Where or, more specifically, the AtWhere) starting at *TP1* (StartWhen, the point in time at which phase 1 starts). It should happen fast (Mod, *in-manner fast*) and has the purpose of enabling that bridge to be secured (the Why). The securing that should be enabled by the occupation of the bridge is a task also assigned to *Coy2*. It is assigned the label *label-o24*. The label for the occupation task itself is *label-o23*.

The most important constituent for our study is the Where. The spatial constraints for the assigned task have to be listed here. In Schade & Hieb (2006a), the Where constituent in a task assignment is either a RouteWhere (in the case that the task in question involves a movement) or an AtWhere (in all the other cases). The basic discovery of this study is that this is not sufficient. The Where constituent of a task assignment has to list all the spatial constraints that apply to the task to be assigned. Only then can a system be enabled to interpret the task assignment correctly from the geospatial view. To a soldier, who is a human expert in interpreting overlays, the spatial constraints are quite easily recognizable from the overlay that is attached to the order which contains the task assignment in question. That overlay, however, includes all spatial constraints for all the task assignments of that order, so that there are two major challenges. First, how can the overlay be represented such that a system can interpret all the graphics on the overlay correctly (i.e., as intended by the tasker who had drawn the overlay), and second, how can a system identify those parts of the overlay that are relevant for a specific task assignment. Solutions to these problems will be presented in the following sections. In the section "Representing Overlays", it will be explained how overlays can be represented formally. In the section "Overlay Excerpts and their Representation", it will be shown how the parts of an overlay relevant for a specific task assignment can be extracted and represented. This representation is called "overlay excerpt" and it always is task-specific.

The formal representations of overlays, referred to below as "general overlays", as well as overlay excerpts have a name, a unique ID, by which they can be referred to. Thus, in order to connect an overlay excerpt to its task assignment in the respective C2LG expression, a new rule has been introduced in C2LG:

(4) Where → ***under-use-of*** OverlayExcerpt

Rule (4) means that the Where of a C2LG task assignment expression can be expanded to a sequence that consists of the keyword "*under-use-of*" and the name (ID) of an overlay excerpt. Since the C2LG is a lexical grammar, the overlay excerpt that is referred to in the Where constituent has to fit to the task itself. In the original tasking grammar (Schade & Hieb, 2006a), the tasking verb determines whether the Where is an AtWhere or a RouteWhere. Using overlay excerpts and rule (4), the connection between the tasking verb and the Where is both more complex and more precise. Overlay excerpts have a type that denotes the task they can be connected to. How this kind of constraint is realized will be explained in detail in the section "Putting Things Together: Assigning Overlay Excerpts to C2LG Expressions". That section also discusses the processing of C2LG expressions, that is, the calculation of the corresponding c-structure and a-structure. First, however, it has to be discussed how to represent overlays and overlay excerpts formally.

Representing Overlays

This section discusses how to represent an overlay formally. This formal representation of an overlay might be called "C2LG overlay". However, in order to avoid confusion with overlay excerpts that also are formal representations associated to C2LG expressions, the formal overlay is called "general overlay" in the following.

A general overlay consists of the following components:

- the overlay's label (its name or its unique ID) for reference,
- a reference to a map including the coordinates of the map's upper left and lower right corners,
- a sequence of units by name and coordinates, and
- a sequence of control feature representations.

While the first three parts are self-explanatory, control feature representations are not. They are presented and discussed in section "Representing Tactical Spatial Objects" which follows section "Overlay Excerpts and their Representations".

In order to be processable, general overlays are written in XML. The XML type of a general overlay which specifies the look of a general overlay according to the list of its components is given in Appendix A. That type specifies that the sequence of represented control features always has to start with the representation of the area of interest for the whole operation ordered by the order the overlay is attached to.

Overlay Excerpts and their Representation

Overlay excerpts correspond to a task assignment. An overlay except consists of

- a name (or ID) for reference,
- the name of the general overlay they belong to,
- the name and the coordinates of the unit the task is assigned to (the taskee),
- names and coordinates of units affected by the task (if any), and
- a sequence of control feature representations.

Overlay excerpts are also written in XML. Since they are referred to in the Where constituent of a task assignment expression, they have to fit to the respective task. Thus, for each kind of task there is a specific XML schema that describes how the overlay excerpt for such a task has to look. However, all these overlay excerpts follow the XML type of an overlay excerpt which reflects the given list of an overlay excerpt's components (also shown in appendix A). The differences among the XML schemata for overlay excerpts lies in the sequences of control feature representation, because for different types of tasks different types of control features are mandatory, optional or forbidden. E.g., a delay tasks demands delay lines and a movement to contact task demands a limit of advance but not vice versa. However, like general overlays, the sequence of control feature representations within an overlay excerpt also always start with the representation of an area in interest, in this case the area in which the respective task has to be executed.

Appendix B contains an example of a schemata for an overlay excerpt, corresponding to the task assignment "establish a Casualty Collection Point" (CCP).

Representing Tactical Spatial Objects

Both the general overlay and the overlay excerpts contain control feature representations. "Control Feature" is the term for non-substantial tactical spatial objects which are used by the Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM). Control features are points, like Control Points, lines, like Phase Lines, or areas, like the Area of Interest.

The formal representation of control features compatible with the C2LG, general overlays, and overlay excerpts consists of

- the label (name or ID) of the control feature,
- the name of the unit that "owns" the feature,
- the name of the unit that uses the feature,
- the feature's geometric type,
- its subtype,
- its defining coordinates, and
- a so-called "part_of" entry.

In addition, a control feature representation may list other control features, namely those that are associated with it, as well as additional attributes. For example, evacuation routes can be listed as associated control features for a CCP.

Control feature representations are again written in XML. There are three types of these representations corresponding to the three geometric types (point, line, and area). The main difference between these types is related to the number of defining coordinates demanded. The type for areas demands at least three, the type for the lines at least two, and the type for points only one. The three XML types for control feature representations of the types area, line, and point are shown in Appendix A.

The label (name or ID) of a control feature is its unique identifier by which one can refer to the feature. The owner of a control feature is the unit which defines it. The user is the unit that makes use of it. Normally, the owner is superior to the user. For example, a battalion staff defines the areas of interest and all their boundaries for the battalion's companies and each of the companies uses its own assigned area (and the respective boundaries). A control feature's geometric type says whether the feature geometrically is a point, a line, or an area. Its subtype is the military denotation of that feature, e.g., an area of interest has subtype "area of interest" and a phase line has subtype "phase line". "Part_of" information tells us whether the feature is part of another feature (cf. below).

In control feature representations, the label is mandatory, but the other elements are optional. However, the following business rule is to be obeyed: If a control feature is listed in an overlay excerpt and if it is also listed in the general overlay to which the excerpt belongs, then the control feature is only to be listed by name in the excerpt. The additional information then has to be taken from the general overlay. If the control feature is listed in an excerpt but not in its general overlay, then the information about its owner, its user, its type, its subtype, and its defining coordinates is requested (and thus mandatory). The label, the owner's name, the user's name, the type, the subtype, and the coordination information is always requested (and thus mandatory) when a control feature is listed in a general overlay.

The "part_of" information is syntactically always optional. It is included if the feature is a part of another feature. For example, there might be a phase line called "PL Chryses" in a general overlay attached to a battalion order. If the three companies of that battalion are tasked to advance to PL Chryses, the companies will have to do this in their own specific area of interest of which each only includes a part of PL Chryses. Then, for example, the section of PL Chryses that serves as phase line for the A company might be called "PL ChrysesA". In this case, the defining entry for PL ChrysesA will include the "part_of" attribute with "PL Chryses" as referring value. With respect to "part_of" the following business rule holds: A feature named as a value of the "part_of" attribute is always referred to exclusively by its label. Otherwise, unwanted recursion creeps into the formalism. The same business rule holds for control features referred to in the association list.

Putting Things Together: Assigning Overlay Excerpts to C2LG Expressions

This section discusses the automatic processing of C2LG expressions and connected geo-information. In principle, the expressions are to be transformed into XML representations which include all the relevant geo-information. The goal is that a system will be able to work with the transformed expression. For example, a simulation system that receives an order in the transformed form should be able to let the respective simulated units execute the tasks assigned by the order as intended. Linguistically, processing an expression that had been generated according to a known grammar means to parse it under exploitation of the knowledge stored in the grammar rules; cf. Carrol (2003) for an overview on computational linguistics' parsing approaches. In our case, the first parsing step transforms an expression into a constituent structure ("c-structure" in the terminology of LFG). Figure 1 shows the tree representation of the c-structure calculated from the C2LG expression "establish MedSupBtl330 MedSupCoy332 CCP332 under-use-of OverlayCCP332 start before 150029FAUG2010 end nlt 170029FAUG2010 in-order-to cause label-ci1 label-o-22;". The expression is a task assignment: The battalion MedSupBtl330 orders its second company MedSupCoy332 to establish a Causality Collection Point (called "CCP331"). The corresponding spatial information is included in the overlay excerpt "OverlayCC332".

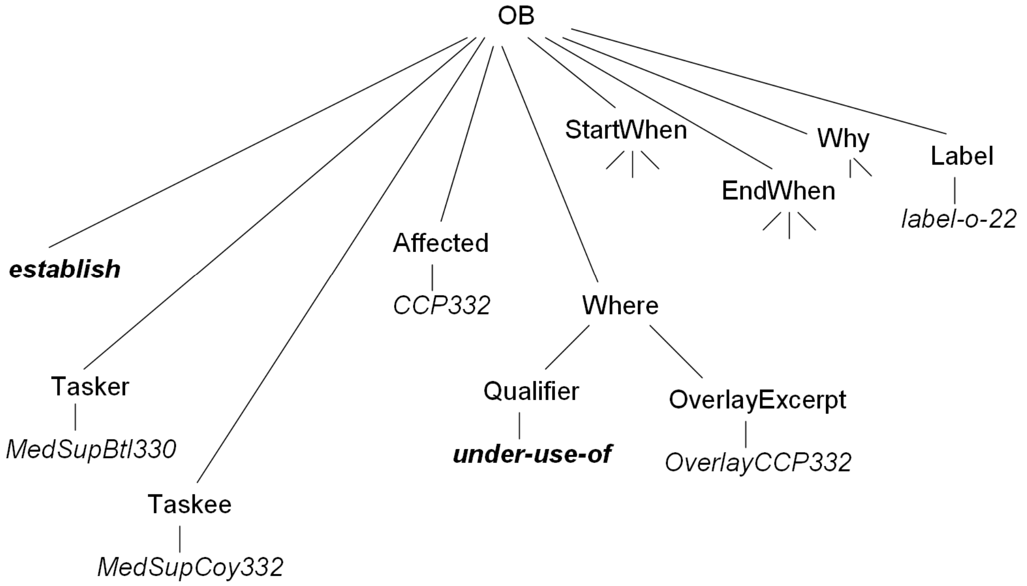
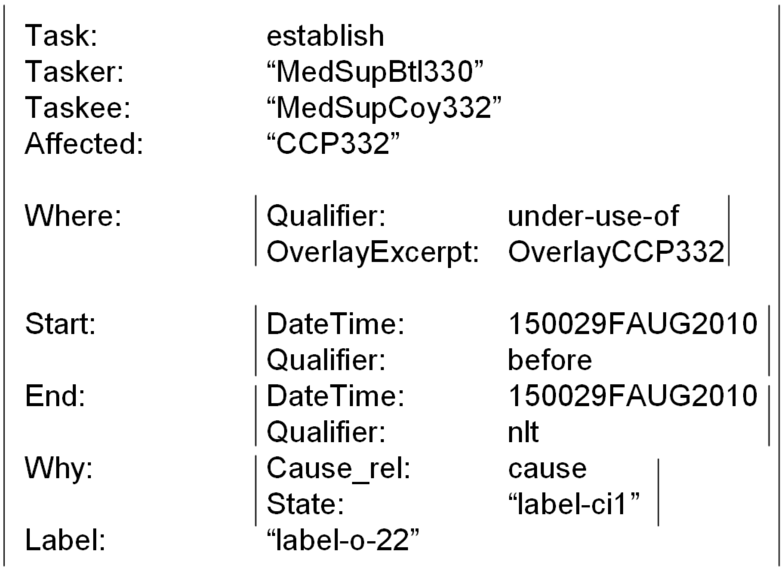


Figure 1: Part of the c-structure for the C2LG expression “establish MedSupBtl330 MedSupCoy332 CCP332 under-use-of OverlayCCP332 start before 150029FAUG2010 end nlt 170029FAUG2010 in-order-to cause label-ci1 label-o-22;”

In the second step, the c-structure is transformed into an argument structure (a-structure in the terminology of LFG). As C2LG generates unambiguous expressions, there are no syntax-semantic gaps as in natural languages. Therefore the argument structure can be derived directly from the constituent structure, whereas in the parsing of natural language expressions by standard LFG-based parsers, a functional structure (f-structure) has to be created before the a-structure can be calculated.



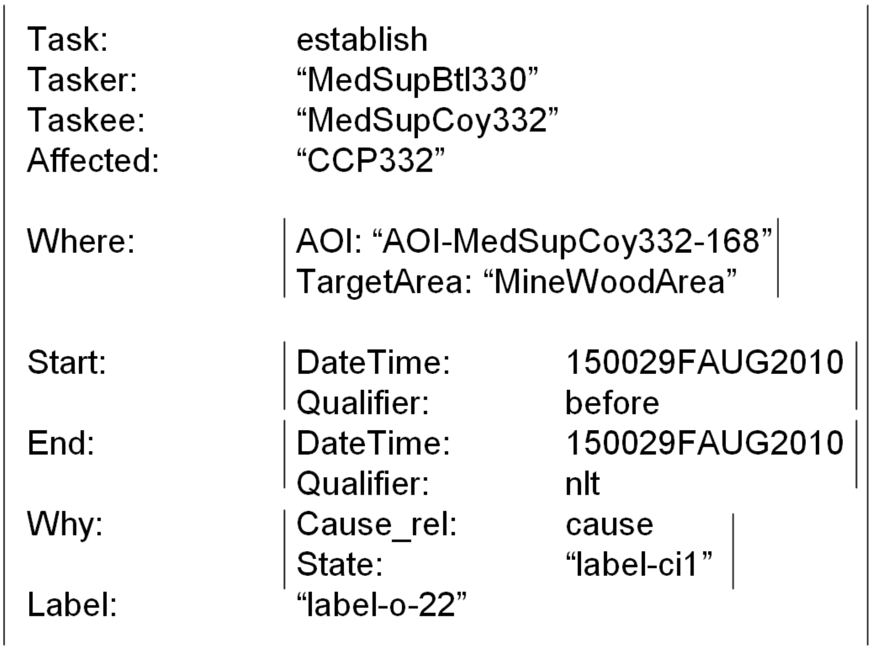
Task:	establish	
Tasker:	“MedSupBtl330”	
Taskee:	“MedSupCoy332”	
Affected:	“CCP332”	
Where:	Qualifier:	under-use-of
	OverlayExcerpt:	OverlayCCP332
Start:	DateTime:	150029FAUG2010
	Qualifier:	before
End:	DateTime:	150029FAUG2010
	Qualifier:	nlt
Why:	Cause_rel:	cause
	State:	“label-ci1”
Label:	“label-o-22”	

Figure 2a: a-structure for the C2LG expression *“establish MedSupBtl330 MedSupCoy332 CCP332 under-use-of OverlayCCP332 start before 150029FAUG2010 end nlt*

170029FAUG2010 in-order-to cause label-ci1 label-o-22;" without resolving the Where attribute

Using C2LG and the developed representations for overlays, overlay excerpts, and tactical spatial objects, the crucial part of the transformation from the constituent structure into the argument structure is resolving the value of the Where attribute. The argument structure for the example task assignment without resolving the Where attribute is shown in figure 2a.

In the constituent structure, the Where value consists of a qualifier and a reference. If the qualifier is the key word "under-use-of", the reference is to an overlay excerpt fitting the task. Resolving a Where value referring to an overlay excerpt means to take all control feature information from the overlay excerpt and calculate an attribute-value pair out of each control feature listed. For each such pair, the attribute is the feature's subtype whereas the value is either its name or its sequence of coordinates. In the example presented in figures 1, 2a, and 2b an overlay excerpt for an "establish" task is analyzed. It has only two control features, the current area of interest of the MedSupCoy332 and the target area where the casualty collection point is to be established. In Figure 2b, the respective control features' names ("AOI-MedSupCoy332-168" and "MineWoodArea") are shown as values. Names can be taken as values if an underlying data base is available that can provide a feature's coordinates when the name is known. Otherwise, respective sequences of coordinates must be used as values.



Task:	establish	
Tasker:	"MedSupBtl330"	
Taskee:	"MedSupCoy332"	
Affected:	"CCP332"	
Where:	AOI: "AOI-MedSupCoy332-168" TargetArea: "MineWoodArea"	
Start:	DateTime:	150029FAUG2010
	Qualifier:	before
End:	DateTime:	150029FAUG2010
	Qualifier:	nlt
Why:	Cause_rel:	cause
	State:	"label-ci1"
Label:	"label-o-22"	

Figure 2b: a-structure for the C2LG expression "*establish MedSupBtl330 MedSupCoy332 CCP332 under-use-of OverlayCCP332 start before 150029FAUG2010 end nlt 170029FAUG2010 in-order-to cause label-ci1 label-o-22*;" after resolving the Where attribute

In a third step, the argument structure is transformed into XML. This is a trivial step. Attributes in the argument structure (attribute-value matrix) are used as XML tags and the values are their content.

In the end, the expression of the order in question is represented in XML including all the necessary geo-information. This geo-information is always represented under the <Where>-tag. The XML representation now can be sent to other systems, such as C2 systems, simulation systems or robotic forces, which are to execute the task assignment.

Lessons Learned and Recommendations

This section lists some of the lessons learned by developing a kind of grammar for geo-information suitable for the C2LG.

First, it is a standard solution to represent information in XML. But in our case, it is more than "standard". Since C2LG is modeled on LFG, C2LG expressions can be parsed to calculate constituent structures (c-structures) and argument structures (a-structures) out of them. In computational linguistics these argument structures are called attribute-value matrices. These matrices can be transformed into XML quite easily by using the attributes as XML tags and the values as content. Thus, there is a well-known parsing process that transforms C2LG expressions into XML, also meaning that geo-information represented in XML can be easily spliced in. This integration of the geo-information is made through the Where constituent that is obligatory in each C2LG task assignment expression. The Where constituent refers to the geo-information relevant for the task assignment in question, represented here as a so-called "overlay excerpt". Then, during the calculation of the argument structure, the XML-coded and relevant geo-information can be taken out of the overlay excerpt and integrated into the argument structure. In the end, it is part of the XML-coded semantic representation of the C2LG expression.

Second, although soldiers can easily read overlays, this is not an easy task for systems. Even if the information represented in an overlay is formally represented, a system has to identify those parts of the overlay information which are relevant for interpreting a single part of an order, e.g., a single task assignment. The linguistic principle of lexicality enforces the development and availability of what is called here "overlay excerpts," which are specific with respect to the task assignments they are attached to. These overlay excerpts can in some part be completed automatically, e.g., the correct reference for the area of interest can be derived from the general overlay. In other parts, the excerpts need to be completed by the user, the one who writes the order. But even in this case, the principle of lexicality combined with the principles of coherence and completeness ensures that the user is asked to provide the relevant geo-information, namely the geo-information that is important for the interpretation of a single task assignment, nothing less and nothing more.

Third, the overlay excerpts are shaped by the Subject Matter Experts (SMEs) since they determine what kind of geo-information is relevant for a specific type of task. However, overlay excerpts can easily be joined together as they follow the simple pattern provided by the XML type of overlay excerpts. Thus, if a new type of task emerges due to advancement in technology or due to change of doctrine (either of our own doctrine or of enemy doctrine), or if the meaning and the circumstances of a certain task change due to the same reasons, not only C2LG task assignment expressions but also the respective overlay excerpts can be easily added or adjusted to the new situation.

Conclusion

The grammar developed inherited the ability of C2LG to assign tasks to units by expressions which are unambiguous and processable by systems. In addition, it allows to connect formally represented geo-information to the task assignments. In order to exploit these qualities of the grammar, geo-information has to be represented formally. Proposals for such representations have been developed. They cover how to represent single spatial objects as well as how to represent whole overlays. The proposed representations are given in XML for processing. Furthermore, they are tuned for integration into the argument structures that result from processing the task assignments expressions.

The developed grammar is a lexical grammar. This ensures that the rules can be fine-tuned by SMEs and thus be adapted to the demands of military doctrine. The formal representations of the tactical spatial objects and the overlays also are laid out in a lexical manner. This not only ensures their compatibility with the language expressions for tasking and reporting but also grants that the grammar's adaptability to doctrinal demands crosses over to the geo-representations.

Future work might consider an automatic transformation from military symbols used in overlays to their formal representations as well as applying the developed methodology to operations other than war or even to civil operations like disaster relief operations.

References

Bresnan, J. (2001). *Lexical-Functional Syntax*. Malden, MA: Blackwell.

Carrol, J. (2003). “Parsing.” In: Mitkov, R. (Ed.), *The Oxford Handbook of Computational Linguistics*. Oxford, UK: Oxford University Press.

Chomsky, N. (1957). *Syntactic Structure*. The Hague, The Netherlands: Mouton.

Hieb, M.R, Kleiner, M., Carey, S. & Schade, U. (2009). Characterizing doctrine through a formalization of C2 processes. Paper presented at the *14th International Command and Control Research and Technology Symposium*. Washington, DC.

Hieb, M.R. & Schade, U. (2007). Formalizing Command Intent Through Development of a Command and Control Grammar. Paper presented at the *12th International Command and Control Research and Technology Symposium*. Newport, RI.

Jurafsky, D. & Martin, J.H. (2009). *Speech and Language Processing*. Upper Saddle River, NJ: Prentice Hall, 2. Edition.

Kunde, D., Orichel, T., Tolk, A., Schade, U., Hieb, M.R. (2009). Harmonizing BML Approaches: Grammars and Data Models for a BML Standard. *2009 Spring Simulation Interoperability Workshop* (09S-SIW-046). San Diego, CA.

Rein, K., Schade, U. & Hieb, M.R. (2009). Battle Management Language (BML) as an Enabler. IEEE International Conference on Communications, *ICC 2009*, Dresden, Germany.

Schade, U. & Hieb, M.R. (2006a). Formalizing Battle Management Language: A Grammar for Specifying Orders. *2006 Spring Simulation Interoperability Workshop* (06S-SIW-068). Huntsville, AL.

Schade, U. & Hieb, M.R. (2006b). Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests, and Reports. Paper presented at the *11th International Command and Control Research and Technology Symposium*, Cambridge, UK.

Schade, U. & Hieb, M.R. (2007). Battle Management Language: A Grammar for Specifying Reports. *2007 Spring Simulation Interoperability Workshop* (07S-SIW-036). Norfolk, VA.

Schade, U. & Hieb, M.R. (2009). Battle Management Language. In: Grosche, J. & Wunder, M. (Eds.), *Verteilte Führungsinformationssysteme*. Heidelberg, Germany: Springer

Schade, U. & Hieb, M.R. (2010). Linguistic Principles of C2 Languages and Grammars. *Spring 2010 Simulation Interoperability Workshop* (10S-SIW-036). Orlando, FL.

Sowa, J.F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks and Cole.

Appendix A: Types

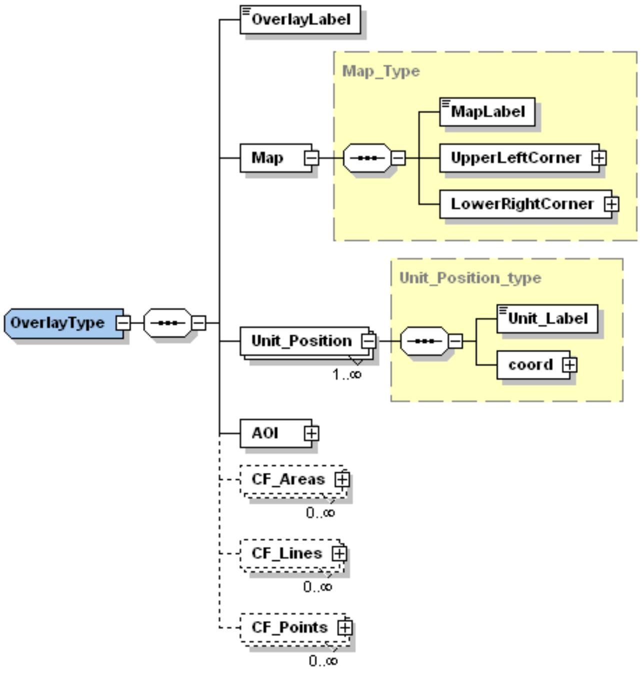


Figure A1: XML type for a general overlay

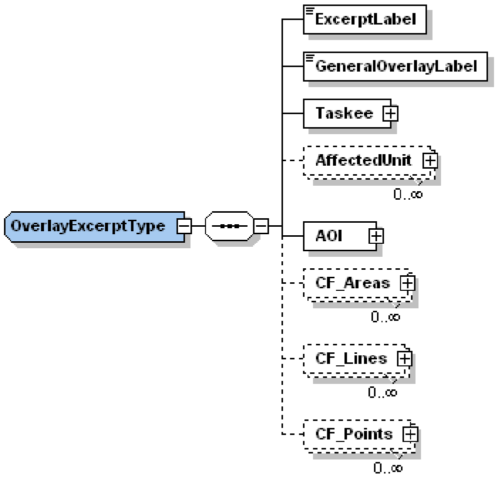


Figure A2: XML type for an overlay excerpt

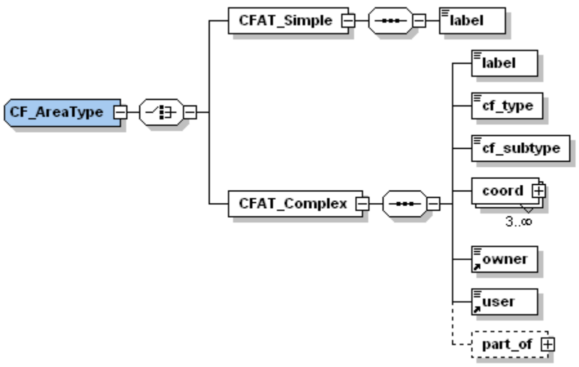


Figure A3: Control feature representation type for control features of type area

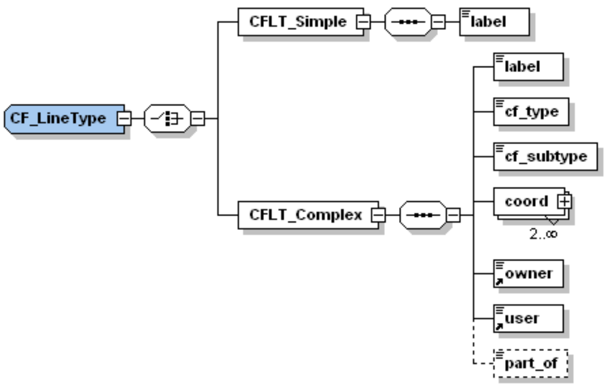


Figure A4: Control feature representation type for control features of type line

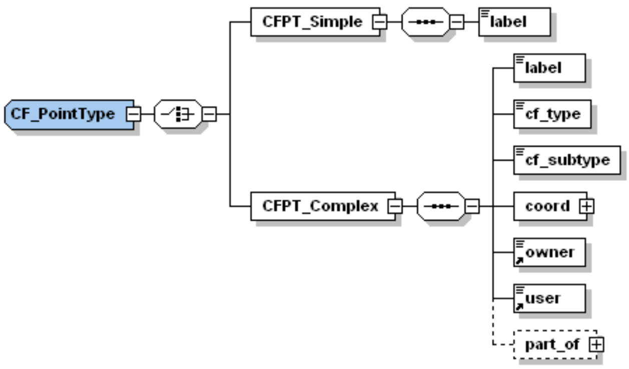


Figure A5: Control feature representation type for control features of type point

Appendix B: XML Schemata

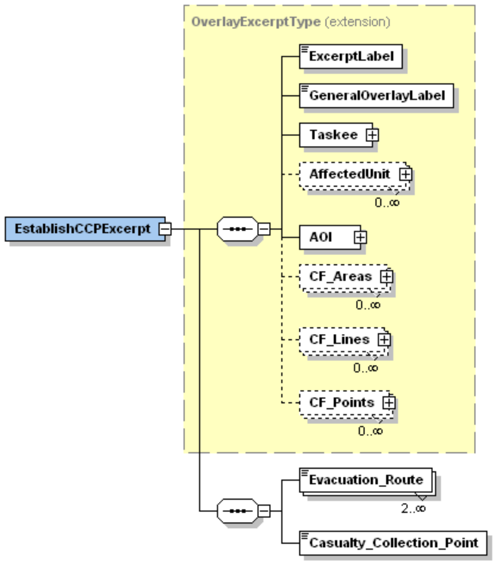


Figure B1: The XML schema for the overlay excerpt to be used for a task of type establishing a CCP. This schema demands references to the CCP as well as to at least two evacuation routes.

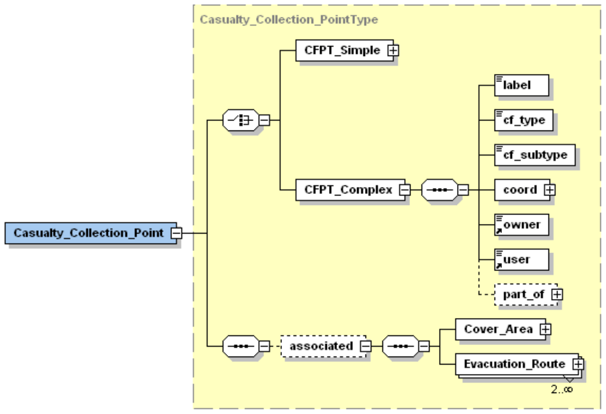


Figure B2: The XML type schema for CCPs. The list of associated control features consists of a cover area and of at least two evacuation routes. Besides, as a CCP is of type “point”, it has only one coordinate.

Appendix C: Example

In the example, the combat battalions (“33”, “99”, and “66”) of brigade XX perform a movement to contact towards phase line Bear (the limit of advance) to secure that phase line. In addition, there is the Medical Support Battalion 330 attached to the brigade to provide medical support. The commander of Medical Support Battalion 330 sends an order to her company *MedSupCoy332*. The Command Intent of that order says that the commander wants to have a Casualty Collection Point (CCP) operational at 1700 of the 29th of August 2010.

Therefore, the company is ordered a) to conduct a tactical road march to the rear of the advancing combat battalions and b) to establish a CCP.

The formal representation of the order in C2LG expressions looks as follows:

[Header]

Sender:	MedSupBtl330
Addressee:	MedSupCoy332
SendingTime:	080029FAUG2010
References:	overlay-MedSupBtl330-168
SecurityClassification:	Unclassified

[Body]

[…]

[Execution]

[Command Intent]
own status-facility 1 own CCP OPR at MedSupBtl330AOI ongoing at 170029FAUG2010 fact label-ci1;

[Task Assignments to the Companies]

```
march MedSupBtl330 MedSupCoy332 under-use-of OverlayExcerptMarch332
        start at  090029FAUG2010 end nlt 150029FAUG2010 in-order-to enable label-o22 label-o21;
establish MedSupBtl330 MedSupCoy332 CCP332 under-use-of OverlayExcerptCCP332
        start before 150029FAUG2010 end nlt 170029FAUG2010 in-order-to cause label-ci1 label-o22;
```

The order refers to an overlay called “overlay-MedSupBtl330-168”. This is represented in the header under the attribute “References” (marked in blue). The task assignments refer to specific overlay excerpts in their respective Where constituents (also marked in blue). These are excerpts of “overlay-MedSupBtl330-168”. The overlay itself (as a simplified sketch) is depicted in figure C1.

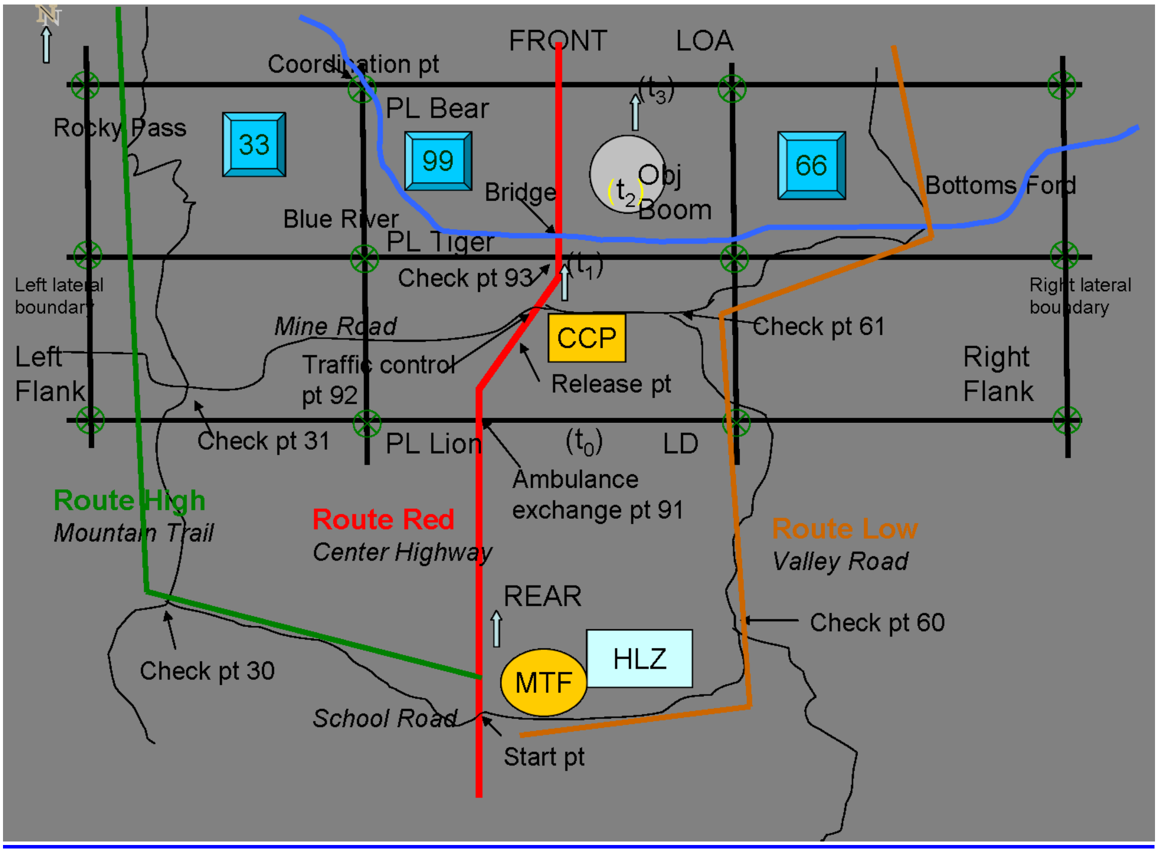


Figure C1: Sketch of “overlay-MedSupBtl330-168”

The formal representation of “overlay-MedSupBtl330-168” is as follows:

```
<GeneralOverlay xsi:noNamespaceSchemaLocation="CCP.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <OverlayLabel>overlay-MedSupBtl330-168</OverlayLabel>
    <Map>
        <MapLabel>map-Janus</MapLabel>
        <UpperLeftCorner><UTMREF>WY850350</UTMREF></UpperLeftCorner>
        <LowerRightCorner><UTMREF>XX550750</UTMREF></LowerRightCorner>
    </Map>
    <Unit_Position>
        <Unit_Label>MedSupBtl330</Unit_Label>
        <coord><UTMREF>XY220170</UTMREF></coord>
    </Unit_Position>
    <AOI>
        <CFAT_Complex>
            <label>AOI-MedSupBtl330-168</label>
            <cf_type>area</cf_type>
            <cf_subtype>AOI</cf_subtype>
            <coord><UTMREF>WY900300</UTMREF></coord>
            <coord><UTMREF>WX900800</UTMREF></coord>
            <coord><UTMREF>XX500800</UTMREF></coord>
            <coord><UTMREF>XY500300</UTMREF></coord>
            <owner>Brigade</owner>
            <user>MedSupBtl330</user>
            <part_of><CFAT_Simple><label>AOI-Brigade-168</label></CFAT_Simple></part_of>
        </CFAT_Complex>
    </AOI>
    <CF_Areas>
        <CFAT_Complex>
            <label>AOI-MedSupCoy331-168</label>
            <cf_type>area</cf_type>
            <cf_subtype>AOI</cf_subtype>
            <coord><UTMREF>WY900300</UTMREF></coord>
            <coord><UTMREF>WX900800</UTMREF></coord>
            <coord><UTMREF>XX100800</UTMREF></coord>
            <coord><UTMREF>XY100300</UTMREF></coord>
```

```
                <owner>MedSupBtl330</owner>
                <user>MedSupCoy331</user>
                <part_of><CFAT_Simple><label>AOI-MedSupBtl330-168</label></CFAT_Simple></part_of>
        </CFAT_Complex>
</CF_Areas>
<CF_Areas>
        <CFAT_Complex>
                <label>AOI-MedSupCoy332-168</label>
                <cf_type>area</cf_type>
                <cf_subtype>AOI</cf_subtype>
                <coord><UTMREF>XY100300</UTMREF></coord>
                <coord><UTMREF>XX100800</UTMREF></coord>
                <coord><UTMREF>XX300800</UTMREF></coord>
                <coord><UTMREF>XY300300</UTMREF></coord>
                <owner>MedSupBtl330</owner>
                <user>MedSupCoy332</user>
                <part_of><CFAT_Simple><label>AOI-MedSupBtl330-168</label></CFAT_Simple></part_of>
        </CFAT_Complex>
</CF_Areas>
<CF_Areas>
        <CFAT_Complex>
                <label>AOI-MedSupCoy333-168</label>
                <cf_type>area</cf_type>
                <cf_subtype>AOI</cf_subtype>
                <coord><UTMREF>XY300300</UTMREF></coord>
                <coord><UTMREF>XX300800</UTMREF></coord>
                <coord><UTMREF>XX500800</UTMREF></coord>
                <coord><UTMREF>XY500300</UTMREF></coord>
                <owner>MedSupBtl330</owner>
                <user>MedSupCoy333</user>
                <part_of><CFAT_Simple><label>AOI-MedSupBtl330-168</label></CFAT_Simple></part_of>
        </CFAT_Complex>
</CF_Areas>
<CF_Areas>
        <CFAT_Complex>
                <label>MineWoodArea</label>
                <cf_type>area</cf_type>
                <cf_subtype>AOC</cf_subtype>
                <coord><UTMREF>XY233185</UTMREF></coord>
                <coord><UTMREF>XY230180</UTMREF></coord>
                <coord><UTMREF>XY240180</UTMREF></coord>
                <coord><UTMREF>XY240185</UTMREF></coord>
                <owner>MedSupBtl330</owner>
                <user>MedSupCoy332</user>
        </CFAT_Complex>
</CF_Areas>
<CF_Areas>
        <CFAT_Complex>
                <label>MTF</label>
                <cf_type>area</cf_type>
                <cf_subtype>AOR</cf_subtype>
                <coord><UTMREF>XX219845</UTMREF></coord>
                <coord><UTMREF>XX219844</UTMREF></coord>
                <coord><UTMREF>XX220844</UTMREF></coord>
                <coord><UTMREF>XX220845</UTMREF></coord>
                <owner>MedSupBtl330</owner>
                <user>MedSupBtl330</user>
        </CFAT_Complex>
</CF_Areas>
<CF_Lines>
        <CFLT_Complex>
                <label>LeftBoundary-MedSupBtl330-168</label>
                <cf_type>line</cf_type>
                <cf_subtype>BDYOR</cf_subtype>
                <coord><UTMREF>WY900300</UTMREF></coord>
                <coord><UTMREF>WX900800</UTMREF></coord>
                <owner>Brigade</owner>
                <user>MedSupBtl330</user>
        </CFLT_Complex>
</CF_Lines>
<CF_Lines>
        <CFLT_Complex>
                <label>RightBoundary-MedSupBtl330-168</label>
                <cf_type>line</cf_type>
                <cf_subtype>BDYOR</cf_subtype>
                <coord><UTMREF>XY500300</UTMREF></coord>
                <coord><UTMREF>XX500800</UTMREF></coord>
        <owner>Brigade</owner>
        <user>MedSupBtl330</user>
    </CFLT_Complex>
</CF_Lines>
<CF_Lines>
    <CFLT_Complex>
        <label>LeftBoundary-MedSupCoy331-168</label>
        <cf_type>line</cf_type>
        <cf_subtype>BDYOR</cf_subtype>
        <coord><UTMREF>WY900300</UTMREF></coord>
        <coord><UTMREF>WX900800</UTMREF></coord>
        <owner>MedSupBtl330</owner>
        <user>MedSupCoy331</user>
        <part_of>
            <CFLT_Simple><label>LeftBoundary-MedSupBtl330-168</label></CFLT_Simple>
        </part_of>
    </CFLT_Complex>
</CF_Lines>
<CF_Lines>
    <CFLT_Complex>
        <label>RightBoundary-MedSupCoy331-168</label>
        <cf_type>line</cf_type>
        <cf_subtype>BDYOR</cf_subtype>
        <coord><UTMREF>XY100300</UTMREF></coord>
        <coord><UTMREF>XX100800</UTMREF></coord>
        <owner>MedSupBtl330</owner>
        <user>MedSupCoy331</user>
    </CFLT_Complex>
</CF_Lines>
<CF_Lines>
    <CFLT_Complex>
        <label>LeftBoundary-MedSupCoy332-168</label>
        <cf_type>line</cf_type>
        <cf_subtype>BDYOR</cf_subtype>
        <coord><UTMREF>XY100300</UTMREF></coord>
        <coord><UTMREF>XX100800</UTMREF></coord>
        <owner>MedSupBtl330</owner>
        <user>MedSupCoy332</user>
    </CFLT_Complex>
</CF_Lines>
<CF_Lines>
    <CFLT_Complex>
        <label>RightBoundary-MedSupCoy332-168</label>
        <cf_type>line</cf_type>
        <cf_subtype>BDYOR</cf_subtype>
        <coord><UTMREF>XY300300</UTMREF></coord>
        <coord><UTMREF>XX300800</UTMREF></coord>
        <owner>MedSupBtl330</owner>
        <user>MedSupCoy332</user>
    </CFLT_Complex>
</CF_Lines>
<CF_Lines>
    <CFLT_Complex>
        <label>LeftBoundary-MedSupCoy333-168</label>
        <cf_type>line</cf_type>
        <cf_subtype>BDYOR</cf_subtype>
        <coord><UTMREF>XY300300</UTMREF></coord>
        <coord><UTMREF>XX300800</UTMREF></coord>
        <owner>MedSupBtl330</owner>
        <user>MedSupCoy333</user>
    </CFLT_Complex>
</CF_Lines>
<CF_Lines>
    <CFLT_Complex>
        <label>RightBoundary-MedSupCoy333-168</label>
        <cf_type>line</cf_type>
        <cf_subtype>BDYOR</cf_subtype>
        <coord><UTMREF>XY500300</UTMREF></coord>
        <coord><UTMREF>XX500800</UTMREF></coord>
        <owner>MedSupBtl330</owner>
        <user>MedSupCoy333</user>
        <part_of>
            <CFLT_Simple><label>RightBoundary-MedSupBtl330-168</label></CFLT_Simple>
        </part_of>
    </CFLT_Complex>
</CF_Lines>
<CF_Lines>
```

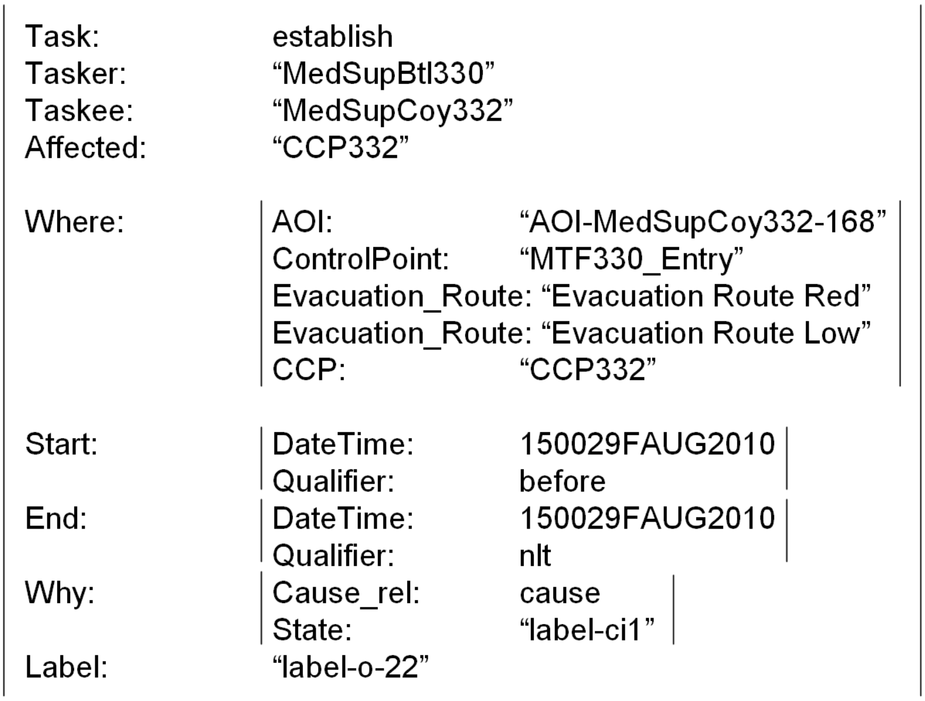
```
        <CFLT_Complex>
                <label>PL Bear</label>
                <cf_type>line</cf_type>
                <cf_subtype>PHLINE</cf_subtype>
                <coord><UTMREF>WY900300</UTMREF></coord>
                <coord><UTMREF>XY500300</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
        </CFLT_Complex>
</CF_Lines>
<CF_Lines>
        <CFLT_Complex>
                <label>LOA Bear</label>
                <cf_type>line</cf_type>
                <cf_subtype>LIMADV</cf_subtype>
                <coord><UTMREF>WY900300</UTMREF></coord>
                <coord><UTMREF>XY500300</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
                <part_of><CFLT_Simple><label>PL Bear</label></CFLT_Simple></part_of>
        </CFLT_Complex>
</CF_Lines>
<CF_Lines>
        <CFLT_Complex>
                <label>PL Tiger</label>
                <cf_type>line</cf_type>
                <cf_subtype>PHLINE</cf_subtype>
                <coord><UTMREF>WY900200</UTMREF></coord>
                <coord><UTMREF>XY500200</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
        </CFLT_Complex>
</CF_Lines>
<CF_Lines>
        <CFLT_Complex>
                <label>PL Lion</label>
                <cf_type>line</cf_type>
                <cf_subtype>PHLINE</cf_subtype>
                <coord><UTMREF>WY900100</UTMREF></coord>
                <coord><UTMREF>XY500100</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
        </CFLT_Complex>
</CF_Lines>
<CF_Lines>
        <CFLT_Complex>
                <label>RouteHigh</label>
                <cf_type>line</cf_type>
                <cf_subtype>ROUTE</cf_subtype>
                <coord><UTMREF>XX200850</UTMREF></coord>
                <coord><UTMREF>WX930900</UTMREF></coord>
                <coord><UTMREF>WY935110</UTMREF></coord>
                <coord><UTMREF>WY920300</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
        </CFLT_Complex>
</CF_Lines>
<CF_Lines>
        <CFLT_Complex>
                <label>RouteRed</label>
                <cf_type>line</cf_type>
                <cf_subtype>ROUTE</cf_subtype>
                <coord><UTMREF>XX200800</UTMREF></coord>
                <coord><UTMREF>XY200100</UTMREF></coord>
                <coord><UTMREF>XY200120</UTMREF></coord>
                <coord><UTMREF>XY230180</UTMREF></coord>
                <coord><UTMREF>XY235190</UTMREF></coord>
                <coord><UTMREF>XY235200</UTMREF></coord>
                <coord><UTMREF>XY235215</UTMREF></coord>
                <coord><UTMREF>XY235300</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
        </CFLT_Complex>
</CF_Lines>
<CF_Lines>
        <CFLT_Complex>
                <label>RouteLow</label>
                <cf_type>line</cf_type>
                <cf_subtype>ROUTE</cf_subtype>
                <coord><UTMREF>XX200830</UTMREF></coord>
                <coord><UTMREF>XX300840</UTMREF></coord>
                <coord><UTMREF>XX300950</UTMREF></coord>
                <coord><UTMREF>XY320100</UTMREF></coord>
                <coord><UTMREF>XY270170</UTMREF></coord>
                <coord><UTMREF>XY420210</UTMREF></coord>
                <coord><UTMREF>XY400300</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
        </CFLT_Complex>
</CF_Lines>
<CF_Points>
        <CFPT_Complex>
                <label>CheckPoint30</label>
                <cf_type>point</cf_type>
                <cf_subtype>CKPGEN</cf_subtype>
                <coord><UTMREF>WX930900</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
                <part_of><CFPT_Simple><label>RouteHigh</label></CFPT_Simple></part_of>
        </CFPT_Complex>
</CF_Points>
<CF_Points>
        <CFPT_Complex>
                <label>CheckPoint31</label>
                <cf_type>point</cf_type>
                <cf_subtype>CKPGEN</cf_subtype>
                <coord><UTMREF>WY935110</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
                <part_of><CFPT_Simple><label>RouteHigh</label></CFPT_Simple></part_of>
        </CFPT_Complex>
</CF_Points>
<CF_Points>
        <CFPT_Complex>
                <label>CheckPoint60</label>
                <cf_type>point</cf_type>
                <cf_subtype>CKPGEN</cf_subtype>
                <coord><UTMREF>XX300950</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
                <part_of><CFPT_Simple><label>RouteLow</label></CFPT_Simple></part_of>
        </CFPT_Complex>
</CF_Points>
<CF_Points>
        <CFPT_Complex>
                <label>CheckPoint61</label>
                <cf_type>point</cf_type>
                <cf_subtype>CKPGEN</cf_subtype>
                <coord><UTMREF>XY270170</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
                <part_of><CFPT_Simple><label>RouteLow</label></CFPT_Simple></part_of>
        </CFPT_Complex>
</CF_Points>
<CF_Points>
        <CFPT_Complex>
                <label>CheckPoint91</label>
                <cf_type>point</cf_type>
                <cf_subtype>CKPGEN</cf_subtype>
                <coord><UTMREF>XY200100</UTMREF></coord>
                <owner>Division</owner>
                <user>Brigade</user>
                <part_of><CFPT_Simple><label>RouteRed</label></CFPT_Simple></part_of>
        </CFPT_Complex>
</CF_Points>
<CF_Points>
        <CFPT_Complex>
                <label>AmbulanceExchangePoint91</label>
                <cf_type>point</cf_type>
                <cf_subtype>RNDZPT</cf_subtype>
                <coord><UTMREF>XY200100</UTMREF></coord>
                <owner>MedSupBtl330</owner>
                <user>MedSupCoy332</user>
                <part_of><CFPT_Simple><label>CheckPoint91</label></CFPT_Simple></part_of>
        </CFPT_Complex>
    </CF_Points>
    <CF_Points>
        <CFPT_Complex>
            <label>CheckPoint92</label>
            <cf_type>point</cf_type>
            <cf_subtype>CKPGEN</cf_subtype>
            <coord><UTMREF>XY230180</UTMREF></coord>
            <owner>Division</owner>
            <user>Brigade</user>
            <part_of><CFPT_Simple><label>RouteRed</label></CFPT_Simple></part_of>
        </CFPT_Complex>
    </CF_Points>
    <CF_Points>
        <CFPT_Complex>
            <label>CheckPoint93</label>
            <cf_type>point</cf_type>
            <cf_subtype>CKPGEN</cf_subtype>
            <coord><UTMREF>XY235200</UTMREF></coord>
            <owner>Division</owner>
            <user>Brigade</user>
            <part_of><CFPT_Simple><label>RouteRed</label></CFPT_Simple></part_of>
        </CFPT_Complex>
    </CF_Points>
</GeneralOverlay>
```

The overlay excerpts referred to in the C2LG task assignments are excerpts of this overlay. For example, the task assignment "*establish MedSupBtl330 MedSupCoy332 CCP332 under-use-of OverlayExcerptCCP332 start before 150029FAUG2010 end nlt 170029FAUG2010 in-order-to cause label-ci1 label-o22;*" refers to overlay excerpt "OverlayExcerptCCP332" which is as follows:

```
<EstablishCCP_Excerpt xsi:noNamespaceSchemaLocation="CCP.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <ExcerptLabel>OverlayExcerptCCP332</ExcerptLabel>
    <GeneralOverlayLabel>overlay-MedSupBtl330-168</GeneralOverlayLabel>
    <AOI>
        <CFAT_Simple><label>AOI-MedSupCoy332-168</label></CFAT_Simple>
    </AOI>
    <CF_Points>
        <CFPT_Complex>
            <label>MTF330_Entry</label>
            <cf_type>point</cf_type>
            <cf_subtype>CTLPNT</cf_subtype>
            <coord><UTMREF>XX200850</UTMREF></coord>
            <owner>MedSupBtl330</owner>
            <user>MedSupCoy332</user>
        </CFPT_Complex>
    </CF_Points>
    <Evacuation_Route>
        <CFLT_Complex>
            <label>Evacuation Route Red</label>
            <cf_type>line</cf_type>
            <cf_subtype>ROUTE</cf_subtype>
            <coord><UTMREF>XX200850</UTMREF></coord>
            <coord><UTMREF>XY200130</UTMREF></coord>
            <coord><UTMREF>XY220150</UTMREF></coord>
            <owner>MedSupBtl330</owner>
            <user>MedSupCoy332</user>
            <part_of><CFLT_Simple><label>Route Red</label></CFLT_Simple></part_of>
        </CFLT_Complex>
        <associated>
            <Traffic_Contol_Point>
                <CFPT_Simple><label>ControlPoint91</label></CFPT_Simple>
            </Traffic_Contol_Point>
            <Patient_DropOff_Point>
                <CFPT_Simple><label>MTF330_Entry</label></CFPT_Simple>
            </Patient_DropOff_Point>
        </associated>
        <additional_attributes>
            <priority>1</priority>
            <usability>high</usability>
            <available>yes</available>
```

```
                <//additional_attributes>
        </Evacuation_Route>
        <Evacuation_Route>
                <CFLT_Complex>
                        <label>Evacuation Route Low</label>
                        <cf_type>line</cf_type>
                        <cf_subtype>ROUTE</cf_subtype>
                        <coord><UTMREF>XX200850</UTMREF></coord>
                        <coord><UTMREF>XX300900</UTMREF></coord>
                        <coord><UTMREF>XY270150</UTMREF></coord>
                        <coord><UTMREF>XY220150</UTMREF></coord>
                        <owner>MedSupBtl330</owner>
                        <user>MedSupCoy332</user>
                        <part_of><CFLT_Simple><label>Route Low</label></CFLT_Simple></part_of>
                </CFLT_Complex>
                <associated>
                        <Traffic_Contol_Point>
                                <CFPT_Simple><label>ControlPoint60</label></CFPT_Simple>
                                <CFPT_Simple><label>ControlPoint61</label></CFPT_Simple>
                        </Traffic_Contol_Point>
                        <Patient_DropOff_Point>
                                <CFPT_Simple><label>MTF330_Entry</label></CFPT_Simple>
                        </Patient_DropOff_Point>
                </associated>
                <additional_attributes>
                        <priority>2</priority>
                        <usability>medium</usability>
                        <available>yes</available>
                </additional_attributes>
        </Evacuation_Route>
        <Casualty_Collection_Point>
                <CFPT_Complex>
                        <label>CCP332</label>
                        <cf_type>point</cf_type>
                        <cf_subtype>CTLPNT</cf_subtype>
                        <coord><UTMREF>XY225219</UTMREF></coord>
                        <owner>MedSupBtl330</owner>
                        <user>MedSupCoy332</user>
                </CFPT_Complex>
                <associated>
                        <Cover_Area>
                                <CFAT_Simple><label>Mine Wood Tree Line</label></CFAT_Simple>
                        </Cover_Area>
                        <Evacuation_Route>
                                <CFAT_Simple><label>Evacuation Route Red</label></CFAT_Simple>
                                <CFAT_Simple><label>Evacuation Route Low</label></CFAT_Simple>
                        </Evacuation_Route>
                </associated>
        </Casualty_Collection_Point>
</EstablishCCP_Excerpt>
```

The process to calculate a-structures from the C2LG task assignments has been presented in section “Putting Things Together”. The discussion in that section used basically the same example as discussed here in appendix C. However, there is one difference. In section “Putting Things Together”, the overlay excerpt corresponding to the assignment of the CCP establishing task lists only two tactical spatial objects, namely the area of interest and a target area, i.e., the area in which the CCP has to be established. That overlay excerpt is the excerpt for a mission command style of ordering. The taskee decides by himself at which specific location (in the target area) he will establish the CCP. However, if the geoBML process has determined optimal locations for CCPs in the area of the operation, the tasker might choose one of those locations for the CCP to be established. Then, in the order that CCP is fixed to that location. This is reflected in the corresponding overlay excerpt. This type of excerpt is given here in the appendix. It lists the following tactical spatial objects: the area of interest, the control point which serves as entry point to the MTF, two evacuation routes, and the CCP itself. In the argument structure that is calculated from the C2LG assignment of the task to establish the CCP (cf. figure C2) this is reflected. All the tactical spatial objects mentioned in the excerpt are listed under the Where constituent.



Task:	establish
Tasker:	“MedSupBtl330”
Taskee:	“MedSupCoy332”
Affected:	“CCP332”
Where:	AOI: “AOI-MedSupCoy332-168” ControlPoint: “MTF330_Entry” Evacuation_Route: “Evacuation Route Red” Evacuation_Route: “Evacuation Route Low” CCP: “CCP332”
Start:	DateTime: 150029FAUG2010 Qualifier: before
End:	DateTime: 150029FAUG2010 Qualifier: nlt
Why:	Cause_rel: cause State: “label-ci1”
Label:	“label-o-22”

Figure C2: the a-structure for the C2LG expression “*establish MedSupBtl330 MedSupCoy332 CCP332 under-use-of OverlayExcerptCCP332 start before 150029FAUG2010 end nlt 170029FAUG2010 in-order-to cause label-ci1 label-o-22;*”